# COMPUTATIONAL SCIENCE IN THE EDUCATIONAL CURRICULUM

**José-Manuel Cabrera-Delgado**

Inspector of Education of Canary Islands, Spain

jcabdel@gmail.com

**DOI**

https://doi.org/10.23824/ase.v0i27.584

**Abstract**

How to incorporate Computer Science (CS) into the basic education curriculum continues to be subject of controversy at the European level. Without there being a defined strategy on behalf of the European Union in this respect, several countries have begun their incorporation showing us the advantages and difficulties of such action. Main elements of CS, such as computational thinking and coding, are already being taught in schools, establishing the need for a curriculum adapted to the ages of the students, training for teachers and enough resources. The purpose of this article, from the knowledge of the experience of these countries, is to respond, or at least to reflect, on the answers to the following questions: what is CS?, what are their main elements?, why is it necessary?, at what age should CS be taught?, what requirements are needed for their incorporation?

**Key words:** computer science; computational thinking; coding; educational curriculum.

## 1. Introduction

Naughton, J. (2012), professor emeritus of technology at the Open University of the United Kingdom, in his article "A manifesto for teaching computer science in the 21st century"[1], published in "The Guardian" in March 2012 and addressed to Michael Gove, at that time Secretary of State for Education of the United Kingdom, asserted that:

> We teach elementary physics to every child, not primarily to train physicists but because each of them lives in a world governed by physical systems. In the same way, every child should learn some computer science from an early age because they live in a world in which computation is ubiquitous. (p.1)

Such a statement can not be disputed, the world of computing surrounds us at the same time as the speed of technological advances derived from the digital revolution, puts us in front what some have come to call the fourth industrial revolution. How the school must respond to these dizzying changes is a controversial subject and a matter of debate. Without strong scientific evidences that guarantee the best way to act, if it's true that some countries have made decisions on this matter by fear that their citizens will fall behind in skills needed in today's and future society.

That CS develops skills such as problem solving, creativity, critical thinking and learning to learn are the arguments used by non-profit organizations, companies, entities and associations that consider their inclusion in school necessary. Likewise, a direct relationship between their early study and the promotion of STEM[2] areas is argued, Areas where the number of students at European level continues to decline and where demand for university graduates in science, technology, engineering and mathematics is expected to increase in Europe in a 13% until 2025 (European Commission, 2014). Similarly, their study is linked as an effective means for the elimination of gender stereotypes that can reduce the gap in these same areas

[1] Naughton, J. (2012). 'A manifesto for teaching computer science in the 21st century'. The Observer, 31 de marzo. Recovered 26/04/2017 from http://www.guardian.co.uk/education/2012/mar/31/manifesto-teaching-ict-education-minister

[2] Science, Technology, Engineering y Mathematics

between men and women. According to data from the OECD[3], In relation to the 2006 PISA tests for 15-year-old students, less than 5% of girls (average value) plan to continue their studies in some engineering or computer science, very far from the 20% of boys (OECD, 2015).

## 2. What is Computer Science (CS)?

For Wing, J. M. (2006), current vice president of Microsoft Research and one of the most influential authors in this field, is “the study of computation - what can be computed and how to compute it” (p.34). Definition used in the communication titled *Computational Thinking* published by the largest educational and scientific association in terms of computing, the ACM (Association for Computing Machinery).

CSTA, K. (12)[4], in order to establish standards for the teaching of CS in basic education in the United States (K-12), define it as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society. " (p.1).

Finally, and as an example of the multiple definitions that are given, *FECYT* (Spanish Foundation for Science and Technology), *Google* and *Everis* in the report entitled *Education in Computer Science in Spain 2015*[5], defines it as "the way computers are designed and how to generate the precise instructions so that it can execute tasks and solve problems "(p.13).

Definitions that must be complemented with its current character of scientific discipline, far from the minority negation that is experienced in the past because of it’s not related with the fundamental laws of nature. For The Royal Society (2012), CS presents a nature of rigorous discipline at the same level as others such as Mathematics or Physics, containing all the characteristics that make a subject to be considered as a scientific discipline: A body of knowledge, a set of rigorous techniques and methods, a way of thinking and working that provides a different

---

[3] The Organisation for Economic Co-operation and Development (OECD)

[4] Computer Science Teachers Association. https://www.csteachers.org/

[5] FECYT, Google y Everis (2016)

perspective than other disciplines, a stable set of concepts, an independent existence of specific technologies that may appear and disappear in a concrete moment.

## 3. What are the main elements of CS?

For Denning, P.J. (2000) the study of CS has two significant parts. The first one has as objective the study of the representation of the information, as well as the tasks to be followed for its efficient processing. The second is to carry out the design using a computer system.

From this division, two are the reference elements that affect each part: computational thinking and coding.

### 3.1. Computational Thinking

Computational thinking has its beginnings in the works of Papert, S. (1980) and its programming language *Logo*. *Logo* allowed educational use of computers to children in a simple way in order to solve common problems. In 2006, the term computational thinking was recovered by Wing, J. M. (2006), who years later redefined it (Wing, J. M., 2011)[6] as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." (p.1).

CSTA (2011) in in collaboration with ISTE[7], seeking a more operational definition, characterized it by a problem-solving process that includes, but is not limited to:

- Formulate problems in a way that allows us to use a computer and other tools to help solve them.
- Organize and analyze data in a logical way.
- Represent data through abstractions such as models and simulations.
- Automate solutions through algorithmic thinking (a series of ordered steps).

---

[6] https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

[7] International Society for Technology in Education http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

- Identify, analyze and implement possible solutions with the goal of achieving the most efficient combination of steps and resources.
- Generalize and transfer this resolution process to a wide variety of problems.

This process of solving problems or thought processes, using one or another definition, has from its origin as concepts, mechanisms and skills for its execution the following:

- The division or decomposition of the problem.

  Solving complex problems by dividing it into simpler parts that can be solved.

- Recognition of models or patterns.

  Identification of similarities and connections, in such a way that the solution of a new problem can be reached from the base of problems already solved previously.

- Abstraction.

  Search for principles, models or patterns which, detected for the solution of a specific problem, can be applied to a generality of problems that share the same essential characteristics.

- Algorithmic thinking.

  Generation of sequences of operations that applied systematically reach the solution to a problem.

**3.2. Coding**

*FECYT*, *Google* y *Everis* (2016) define coding as: "The process of developing and implementing instructions in a way that allows a computer to perform a task, solve a problem and allow interaction with humans" (p.13).

In recent times, the expression skills that are provided by coding have been repeatedly emphasized. Mitch Resnick, creator of the Scratch[8] programming language, in his famous TED talk[9]: "Let's teach kids to code"[10] makes an analogy between reading and writing and coding. For Resnick (2012) young people have a lot of experience and confidence interacting with new technologies, but a lot less creating and expressing something with them. It's almost as if they could read, but not write with the new technologies.

> When you learn to read and write, it opens up opportunities for you to learn so many other things. When you learn to read, you can then read to learn. And it's the same thing with coding. If you learn to code, you can code to learn. Now some of the things you can learn are sort of obvious. You learn more about how computers work. But that's just where it starts. When you learn to code, it opens up for you to learn many other things. (Resnick, 2012)

For Resnick it's necessary a new way of relation of the technology with the boys and girls of our time. Something that Papert, S. (1980) had already anticipated us before: "children should be programming the computer rather than being programmed by it" (p.5).

### 4. Why is it necessary to include CS in schools?

In the absence of studies that determine a solid scientific basis that prove its convenience, two reasons are argued like relevant to its immediate incorporation into compulsory education.

The first is that CS helps students to acquire competencies that are necessary in current society. Among others, the key competences for lifelong learning of the Recommendation 2006/962/EC, of the European Parliament and of the Council of 18 December 2006. In this respect, it should be noted that although sometimes CS

---

[8] https://scratch.mit.edu/

[9] TED is a non-profit organization dedicated to disseminating ideas, usually in the form of talks (https://www.ted.com)

[10] https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=es

appears related to one of them, specifically, digital competence, it is essential to unlink their absolute equivalence with the following two assertions:

- CS makes it possible to acquire several key competences for learning, not circumscribing to any specific competence. Through computational thinking and programming students must acquire a new form of expression and communication, must improve and use new modes of reasoning for problem solving, use learning as a means of improvement to achieve results, use cooperative work to overcome difficulties, in short, skills that are an integral part of several key competencies.

- Although CS makes it possible to acquire digital competence, such competence requires other knowledge, skills and attitudes that are not included in it. To this end, it is necessary, as has been shown by various studies and reports[11], to differentiate CS from digital literacy (digital skills) and from ICT (Information and Communication Technologies). Distinction that is reflected in the DIGCOMP[12] project, launched by the JRC (Joint Research Centre)[13], with a view to drawing up a common frame of reference at european level that determine it must be understood as a digitally competent citizen. Project that in its last version 2.0 (2016) establishes five main areas of competence to be developed:

    - Information and data literacy
    - Communication and collaboration
    - Digital content creation
    - Safety
    - Problem solving

---

[11] Among others, "INTEF (2016). *El Pensamiento Computacional en la Enseñanza Obligatoria (Computhink)*". Study designed and funded by the Joint Research Centre of the European Commission, carried out by the Institute for Educational Technology of the Italian National Research Council (ITD-CNR) and the European Schoolnet.

[12] Ferrari, A. (2013). DIGCOMP: A framework for developing and understanding digital competence in Europe. Recovered en abril de 2017 de https://www.openeducationeuropa.eu/en/article/DIGCOMP%3A-a-Framework-for-Developing-and-Understanding-Digital-Competence-in-Europe

[13] Center conducting research in order to provide independent scientific advice and support to the policies of the European Union.

A detailed study of these areas and their content allows to delimit what skills, knowledge and attitudes are directly related to CS, that others with the use of technological tools of efficient form (digital literacy), and finally, which of them to the technologies of the Information and communication. Noting that CS, in addition to participating in the adquisition of other competences, does not cover all the content and capabilities of the digital competence.

The second reason is its strength as an element of equity to reduce inequalities of gender, race, social origin or economic level. Regarding gender, multiple studies have confirmed the gap existing between male and female students in scientific careers and CS. And among other actuations to eliminate the stereotypes that make it difficult for women to take an interest in these studies, since 2000[14], not only has been expressed to the convenience of including CS in the educational curriculum in areas related to engineering or science, but also to others such as music and art. Since then, multiple international initiatives aimed at bridging this gap and have asserted that the most effective and impactful place to access all students without differentiation is the school at its elementary educational levels.

**5. At what age should be incorporated CS into school?**

The current trend is that the incorporation of CS to the school should be made from an early age. Even the above, minority sectors contradict this opinion by arguing that it is necessary for the child to have reached a sufficient capacity for abstraction so that this incorporation will be effective. Reductionist trend that forgets the complexity of skills that are put into operation in computational thinking or in coding. For Zapata-Ros, M. (2015) we are faced with a complex competence or rather a complex of skills of which they participate, among others: the bottom-up analysis, the top-down analysis, the heuristics, the divergent thinking, the creativity, the problem solving, the abstract thinking, the recursion, the iteration, the methods by successive approximations, the collaborative methods, patterns, synectics, the metacognition.

[14] American Association of University Women. Educational Foundation. Commission on Technology, Gender, & Teacher Education. (2000). *Tech-Savvy: educating girls in the new computer age*. American Association of University Women.

Different, and at the same time more current debate, is another that directs its attention to how to incorporate CCs. Primarily, whether its inclusion should be performed as an additional area or matter or whether it should be integrated into existing ones, or even whether it should be optional or compulsory. With regard to the European Union, there is no homogeneous response to how it should be included, far from the unanimity that seems to exist in relation to the need for CS to be present in the educational curriculum. In Europe, the majority of countries that have revised their curriculum in recent years have promoted the study of CS in compulsory education in addition to those who propose to introduce it in the near future. England has incorporated it as an additional compulsory subject that all students will study until the end of compulsory education. France has done so recently (2014) in the elementary stage of optional way and extracurricularmente with view to extend it in the stage of secondary (*collège* and *lycée*). Finland, seems to be directed towards a more interdisciplinary work where certain competences are developed by all students. The new Finnish curriculum incorporates programming as a transversal compulsory curricular subject from the first school year.

In our country it doesn't observe a defined strategy for the incorporation of CS in the educational curriculum. Moreover, the fragmentation and high number of areas and subjects seems to condition such decisions, leaving the decision at the mercy of each of the different regional education administrations, which, to incorporate them as additional subjects or areas only have the possibility to do it, either through the block of subjects of free regional configuration, either by complementing the established basic curriculum.

As more representative cases of our country we have to mention to Navarre and Madrid. Navarra has chosen to introduce contents of computer programming in the area of mathematics in the fourth and fifth courses of primary education. For its part, the Community of Madrid has included elements of CS, through the subject "Technology, Programming and Robotics", in the first three courses of secondary education.

**6. What requirements does CS demand for their incorporation at school?**

In order for the inclusion of CS to meet the current needs of our society, based on the previous experiences of the countries mentioned, the following factors seems determinant:

- Improvement of teacher training in CS.
- Design of a progressive curriculum adapted to the different ages.
- The provision of sufficient technical and curricular resources to the teaching staff.

In relation to teacher training, in our country the Common Frame of Digital Competence Teaching[15] presents the following five areas with competencies and associated achievement descriptors:

- Information and data literacy
- Communication and collaboration
- Digital content creation
- Safety
- Problem solving

Although all of them have an impact on the training needed to deliver CS in school, both the digital content creation area and the problem solving area are critical. Especially when one of the competences of the area of creation of digital contents makes direct reference to the coding in the following terms: “Make modifications to computer programs, applications, configurations, programs, devices, understand the principles of the coding, understand what there is behind a program”.

In this regard, It is to highlight the initiative of the Department of Education of the Government of Navarre in the creation of a virtual space dedicated to learning programming, robotics and other emerging technologies called "Code 21"[16], where it collaborates with the Department of Education the Public University of Navarra and

[15] Common Frame of Digital Competence Teaching was born in the year 2012 with the intention of being a descriptive reference for purposes of training, evaluation, accreditation and certification for teachers in relation to digital competence. Coordinated by the National Institute of Educational Technologies and Teacher Training (INTEF) and with the collaboration of the Autonomous Communities and external consultants, it has been updated in 2017.

[16] http://codigo21.educacion.navarra.es/

the Planetarium of Pamplona. Virtual space where multiple resources related to the teaching of these technologies are made available to the educational community, especially those related to primary school teachers.

Beyond our borders, most of the countries that have incorporated CS into primary education have established a specific and individualized training plan for this teaching staff. Many of them, with the collaboration of multinational companies that have provided technical and personal resources so that it could be carried out. In spite of the Navarre initiative, we can say that our country is still far from maintaining a cooperative relationship with technological companies such as that established by countries such as England or the United States. Something that should certainly be the object of reflection and improvement as the educational administrations are one of their best clients.

On the other hand, it is essential that the training of primary school teachers be adapted to the objectives set for this educational stage. Objectives already established by countries like England or the United States that can serve as reference to the countries that plan to incorporate the CS in their curriculum. In England, the objectives of the first two levels (KS1, KS2), which cover the ages of 5 to 11, are aimed at that children being able to create and debug small programs, understand algorithms that respond to simple problems, and know the main programming structures (selection, repetition) that allow them to write small computer programs. In the United States, the objective for the educational primary level (K-6) for the K-12 standards[17] established by CSTA, K. (12), is as follows:

> Elementary school students are introduced to foundational concepts in computer science by integrating basic skills in technology with simple ideas about computational thinking. The learning experiences created from these standards should be inspiring and engaging, helping students see computing as an important part of their world. They should be designed with a focus on active learning, creativity, and exploration and will often be embedded within

[17] https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_Spanish_version.pdf

other curricular areas such as social science, language arts, mathematics, and science. (p.8)

In the same way and given the simplicity required by these ages, it has been generalized as operational tool to work CS, the use of visual programming languages where in addition to acquire basic knowledge, children are encouraged to motivation and creativity attending to work with images and music, mainly. In this sense, programming environments like Alice[18], Greenfoot[19] and in particular Scratch[20] are widely used, allowing personal projects such as creating games, telling stories and making animations.

As for secondary education, it is observed that most of the countries that have opted for the incorporation of CS in the curriculum, either have opted for inclusion as a differentiated subject, or as a complement to subjects related to STEM areas where the teachers have had previous experience with them. This is mainly due to the greater complexity required by the objectives. In particular, in this educational stage the objectives are aimed to allow students to be able to use two or more programming languages, understand complex search and ordering algorithms, use data structures, solve problems through their analysis, design computational solutions, develop the creative capacity, in short, objectives that require the teachers to be destined to its teaching to have a solid formation in CS.

This specialized training, whether it is for primary or secondary education, will inevitably require the design of training structures to achieve it. In this sense, the collaboration of companies, institutions, associations and especially universities, seems fundamental and necessary in both initial training and continuing training. Countries with recent experience in their incorporation have assumed such a need.

---

[18] https://www.alice.org

[19] https://www.greenfoot.org/

[20] https://scratch.mit.edu/

In England, Computing at School[21] is an association where its members, faculty of more than three hundred and fifty colleges, as well as university teachers, educational consultants and representatives of the industrial and professional world, participate voluntarily and freely with the aim of promoting and supporting CS. Computing at School is a project funded by renowned companies like Microsoft, Google, BT. On the other hand, and in consideration of more specialized training according to the educational stage, it has emerged the Barefoot Computing[22] project, which, with the objective of supporting primary education and the creation of adapted resources, has led to the creation of Barefoot communities, communities composed of groups of teachers who share ideas and good practices in relation to the teaching of CS.

In France, the “Teaching computer science in France (Tomorrow can't wait)”[23] report of the Académie des Sciences dedicates a special section in the same to the training of teachers considering it an absolute priority, while establishing a differentiation between the initial formation of the future teachers and the training of the ones who are in active service. Similarly, it differentiates the training to be received by the teachers from the primary stage, from the training to be received by secondary school teachers.

As a last significant example, we should mention to the United States and the organization *Code.org*[24], a non-profit organization dedicated to expanding access to computers, famous for organizing the world event "The Code Hour"[25] that reaches tens of millions of students in more than one hundred and eighty countries. The organization is funded by multiple foundations and technology companies: *Google*, *Facebook*, *Microsoft*. Regarding to trainning teachers, according to its annual report for 2016[26], *Code.org* participated in the preparation of fifty-two thousand teachers in

---

[21] https://www.computingatschool.org.uk/

[22] http://barefootcas.org.uk

[23] Académie des Sciences (2013)

[24] https://code.org/

[25] https://hourofcode.com/es

[26] https://code.org/about/2016

the CS (K-12) standards and established support policies for the promotion of CS in thirty-one of the fifty states.

## Conclusions

Pending common strategies on behalf of the European Union and our country in relation to the incorporation of CS in the educational curriculum, countries that have already begun to implement it in the school begin to show factors that must be considered so that it produces valuable results.

In the first place, teacher training seems to be clearly one of the main elements to be take a care of. Training that must be accommodated to the decisions that must be taken by the different educational administrations in relation to CS. Therefore, if primary education is chosen to follow the path taken by most countries of incorporate CS into the curriculum of other areas, this decision inevitably implies widespread training for all present and future teachers, which will require appropriate training structures for this purpose. In this sense, the decision of an initial formation in CS for the teachers which at the moment is in the Faculties of Education, does not seem to admit more delay. Training should include the above, basic concepts of computational thinking and the use of visual programming languages that allow children to develop in their classrooms the skills and creativity that is provided by CS.

Secondly, it is necessary in both primary and secondary schools a curriculum that clearly establishes the objectives that are proposed for each stage, as well as the correct methodology to achieve them. Thus, if in primary education the option seems to be advanced by an interdisciplinary work of several curricular areas in which the essential elements of CS are worked, in secondary, the decisions that are being taken seem to repeat mistakes that already should have been overcomed. We can not again fall into the mistake of equating CS with digital literacy, more or less advanced. CS must have its own space, within other subjects or with additional subject, without diluting its objectives and contents with others aimed at students being advanced users in the use of information technologies or communication, or experts users of computer applications.

Finally, once established the above, it is necessary that both teachers and students can have enough quality resources so that the expected learning of CS is in accordance to the objectives set. Nowdays, the amount of resources is inexhaustible and continues to increase. There are resources that allow CS to work with computer and resources, called "unplugged", where it is not necessary to use it because it works mainly the computational thinking. Even this, the choice, according to the levels and stages to which it is directed, is not a minor question. Poor decisions can lead to the goals not being reached and, consequently, to the disappointment of the boys and girls and of the teachers themselves. In this sense, as with training, teachers's meetings, either with experts, advisors, or in community, in order to discuss and reflect on the planning and selection of resources, is concretized as one of the best options to consider.

**Financing**

Without express financing.

**Conflict of interests**

None.

**Bibliographic references**

Académie des Sciences. (2013). Teaching computer science in France. Tomorrow can't wait. Recovered in April 2017 from http://www.academie-sciences.fr/pdf/rapport/rads_0513gb.pdf

American Association of University Women. Educational Foundation. Commission on Technology, Gender, & Teacher Education. (2000). *Tech-Savvy: educating girls in the new computer age*. American Association of University Women. Recovered in April 2017 from http://history.aauw.org/files/2013/01/TechSavvy.pdf

Balanskat, A., & Engelhart, K. (2014). Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe. European Schoolnet. Recovered in April 2017 from http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education-Implications for policy and practice* (No. JRC104188). Joint Research Centre (Seville site).

CSTA. (2011). Operational definition of computational thinking. Recovered in April 2017 from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

CSTA, K. (12). Computer science standards. *Computer Science Teachers Association*. Recovered in April 2017 from http://www.education2020.ca/Content/K-12ModelCurrRevEd.pdf

Cabrera, J. M. (2015). Programación informática y robótica en la enseñanza básica. *Avances en Supervisión Educativa*, (24). Recovered in April 2017 from https://avances.adide.org/index.php/ase/article/view/17

Denning, P. J. (2000). Computer Science: The Discipline in Encyclopaedia of Computer Science, Ralston, A., and Hemmendinger. *George Mason University, Fairfax VA*.

Department of Education (2013). Computing programmes of study: key stages 1 and 2. National curriculum in England. Recovered in April 2017 from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf

Department of Education (2013). Computing programmes of study: key stages 3 and 4. National curriculum in England. Recovered in April 2017 from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239067/SECONDARY_national_curriculum_-_Computing.pdf

European Commission. (2014). EU Skills Panorama Analytical Highlight Science, technology, engineering and mathematics (STEM) skills. Recovered in April 2017 from http://skillspanorama.cedefop.europa.eu/sites/default/files/EUSP_AH_STEM_0.pdf

FECYT, Google y Everis (2016). Educación en Ciencias de la Computación en España. Recovered in April 2017 from https://www.fecyt.es/es/publicacion/educacion-de-las-ciencias-de-la-computacion-en-espana

Ferrari, A. (2013). DIGCOMP: A framework for developing and understanding digital competence in Europe. Recovered in April 2017 from https://www.openeducationeuropa.eu/en/article/DIGCOMP%3A-a-Framework-for-Developing-and-Understanding-Digital-Competence-in-Europe

INTEF (2016). El Pensamiento Computacional en la Enseñanza Obligatoria (Computhink). Recovered in April 2017 from http://blog.educalab.es/intef/wp-content/uploads/sites/4/2017/02/2017_0206_CompuThink_JRC_UE-INTEF.pdf

Microsoft America Latina (2016). Enseñar Ciencias de la Computación: Creando oportunidades para los jóvenes de América Latina. Recovered in April 2017 from https://www.yopuedoprogramar.com/CS_Whiter_Paper_Latam.pdf

Naughton, J. (2012). 'A manifesto for teaching computer science in the 21st century'. *The Observer, 31 de marzo*. Recovered 26/04/2017 from http://www.guardian.co.uk/education/2012/mar/31/manifesto-teaching-ict-education-minister

OCDE. (2015). The ABC of Gender Equality in Education: Aptitude, Behaviour, Confidence. OECD. Recovered 26/04/2017 from https://www.oecd.org/pisa/keyfindings/pisa-2012-results-gender-eng.pdf

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..

Roberts, E. S., Kassianidou, M., & Irani, L. (2002). Encouraging women in computer science. ACM SIGCSE Bulletin, 34(2), 84-88. Recovered in April 2017 from http://www-cs.stanford.edu/people/eroberts/papers/SIGCSE-Inroads/EncouragingWomenInCS.pdf

The Royal Society (2012). Shut down or restart? The way forward for computing in UK schools. Recovered in April 2017 from https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

Vuorikari, R., Punie, Y., Gomez, S. C., & Van Den Brande, G. (2016). *DigComp 2.0: The Digital Competence Framework for Citizens. Update Phase 1: The*

*Conceptual Reference Model* (No. JRC101254). Institute for Prospective Technological Studies, Joint Research Centre. Recovered in April 2017 from http://publications.jrc.ec.europa.eu/repository/handle/JRC101254

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. Recovered in April 2017 from http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf

Wing, J. M. (2011). Research Notebook: Computational Thinking - What and Why? *The Link*. Pittsburgh, PA: Carneige Mellon. Página 20. Recovered in April 2017 from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Zapata-Ros, M.(2015). Pensamiento computacional: Una nueva alfabetización digital. RED. *Revista de Educación a Distancia, 46 (4)*. 15 de Septiembre de 2015. Recovered in April 2017 from http://www.um.es/ead/red/46/zapata.pdf

**Normative references**

Decreto Foral 60/2014, de 16 de julio, por el que se establece el currículo de las enseñanzas de Educación primaria en la Comunidad Foral de Navarra. Publicado en BON n. 174 de 05 de Septiembre de 2014.

Decreto 48/2015, de 14 de mayo, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo de la Educación Secundaria Obligatoria.